IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Anand Shridhar Sawant, et al.	§ § §	Confirmation No.:	4997
Serial No.:	10/710,998	§ §	Group Art Unit:	2168
Filed:	08/16/2004	§ §	Examiner:	Jay A. Morrison
For:	File System for Digital Processing Systems with Limited Resources	§ § §	Docket No.:	TI-36864

APPEAL BRIEF

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

Appellants hereby submit this Appeal Brief in connection with the above-identified application. A Notice of Appeal was filed on January 22, 2010.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is Texas Instruments Incorporated of Dallas, Texas, a Delaware corporation.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to Appellant, the Appellant's legal representative, or assignee which will directly affect or be affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF THE CLAIMS

Claims 29-52 are presently pending and rejected. Claims 1-28 are canceled. Claims 29-52 are presently appealed.

IV. STATUS OF THE AMENDMENTS

Claims 29 and 37 were amended pursuant to 37 CFR §1.16 after the final office action dated October 22, 2009. The amendments were entered for purposes of appeal in an Advisory Action dated December 10, 2009.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

All page and line numbers cited below are in reference to the patent specification as downloaded from PAIR. As stored in PAIR, the specification has very large text and no page numbers. Appellants have assigned page numbers to the pages and diligently counted lines as required by the MPEP. However, for convenience and to possibly avoid confusion, Appellants are also providing paragraph numbers from the specification in the citations.

Appellants' disclosure describes methods and systems that enables file systems for applications such as audio/video players to be implemented with limited memory and with fast access to files. *See*, *e.g.*, p. 5, l. 22 – p. 7, l. 12 (paras. [0023] - [0029]). In typical file systems, a file allocation table (FAT) stored on the same nonvolatile storage medium as the file system is used to indicate the cluster identifiers of clusters assigned to files. *See*, *e.g.* p. 2, ll. 6 – 23 (paras. [0006] - [0008]). Further, operations on files such as read, write, and delete access the FAT. *See*, *e.g.*, p. 3, ll. 1-6 (para. [0009]). To speed up such operations, a prior art system described in U.S. Patent No. 6,567,887 issued to Tracy Harmer stores the entire FAT table in RAM, along with the partition table and the directory structures. *See*, *e.g.*, p. 3, ll. 6-13 (para. [0010]). However, some systems such as embedded systems do not have sufficient RAM to support such a storage requirement. *See*, *e.g.*, p. 3, ll. 14-19 (para. [0011]).

In the methods and systems disclosed by Appellants, rather than storing the entire FAT in RAM as in Harmer, a file metadata processing module implements a file open operation that traverses the FAT stored on the file system nonvolatile storage medium and stores only the cluster identifiers of the sequence of clusters allocated to the file in RAM. *See*, *e.g.*, Fig. 7, elements 710, 730; p. 28, l. 23 –p. 29, l. 21; p. 30, ll. 7-11(paras. [0091] – [0093], [0095]). The cluster identifiers are stored in the RAM such that each one is locatable by an index computed using a cluster size and a start offset of data in the file. *See*, *e.g.*, Fig. 4, p. 21, ll. 11-17 (para. [0066]). To read or write data from/to an opened file, the file metadata processing module is overlaid at least partially in memory by a file data processing module (thus reducing the amount of memory required for the file system management software) that implements the file read and write operations. *See*, *e.g.*, Fig. 7, element 720, 740; p. 29, l. 22 – p. 30, l. 6; p. 30, ll. 12- 21 (paras. [0094], [0096]). These read and write operations then use the cluster identifiers stored in RAM to access the file content rather than accessing the FAT on

the nonvolatile storage medium, thus increasing the speed of these operations *See*, *e.g.*, p. 30, ll. 12- 21 (para. [0096]).

Claim 29 is directed to a method for accessing a file in a file system in a protected area comprised in secondary storage of a digital processing system comprising a secure random access memory (RAM). *See*, *e.g.*, Fig. 1, elements 100, 132, 150; p. 8, l. 17 – p. 16, l. 11 (paras. [0032] – [0053]). The method includes opening the file using a file open operation comprised in a file metadata processing module loaded in a shared execution portion of the secure RAM, wherein the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file and stores a cluster identifier for each cluster in the sequence in a buffer comprised in a shared data portion of the secure RAM, wherein the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file. *See*, *e.g.*, Fig. 3, elements 340, 370; Fig. 7, elements 710, 730; p. 19, l. 11 – p. 22, l. 13. (paras. [0060] – [0068]; p. 28, l. 15 – p. 29, l. 21 (paras. [0090] – [0093]). The method also includes accessing the file using a file access operation comprised in a file data processing module loaded in the shared execution portion, wherein the data processing module overlays at least a portion of the metadata processing module, and wherein the file access operation accesses a portion of data in the file using at least one cluster identifier stored in the buffer. *See*, *e.g.*, Fig. 7, elements 720, 740; p. 29, l. 30 – p. 30, l. 21 (paras. [0094] – [0096]).

Claim 37 is directed to a machine readable non-volatile storage medium comprising executable instructions that, when executed by a processor of a digital processing system, cause performance of a method for accessing a file in a file system in a protected area comprised in secondary storage of the digital processing system. *See*, *e.g.*, Fig. 1, elements 100, 132, 150; p. 8, l. 17 – p. 16, l. 11 (paras. [0032] – [0053]); p. 30, l. 22 – p. 32, l. 3 (paras. [0097] – [0100]). The method includes opening the file using a file open operation comprised in a file metadata processing module loaded in a shared execution portion of the secure RAM, wherein the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file and stores a cluster identifier for each cluster in the sequence in a buffer comprised in a shared data portion of the secure RAM, wherein the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file.

See, *e.g.*, Fig. 3, elements 340, 370; Fig. 7, elements 710, 730; p. 19, l. 11 – p. 22, l. 13. (paras. [0060] – [0068]; p. 28, l. 15 – p. 29, l. 21 (paras. [0090] – [0093]). The method also includes accessing the file using a file access operation comprised in a file data processing module loaded in the shared execution portion, wherein the data processing module overlays at least a portion of the metadata processing module, and wherein the file access operation accesses a portion of data in the file using at least one cluster identifier stored in the buffer. *See*, *e.g.*, Fig. 7, elements 720, 740; p. 29, l. 30 – p. 30, l. 21 (paras. [0094] – [0096]).

Claim 45 is directed to a digital processing system that includes a secondary storage comprising a file system in a protected area, wherein the file system comprises a plurality of files. *See*, *e.g.*, Fig. 1, elements 100, 150; p. 8, l. 17 – p. 16, l. 11 (paras. [0032] – [0053]). The digital processing system also includes a secondary storage comprising a file metadata processing module comprising a file open operation and a file data processing module comprising a file access operation. *See*, *e.g.*, Fig. 1, element 170; p. 11, ll. 11-15 (para [0040]); Fig. 7, elements 710, 720; p. 19, p. 28, l. 15 – p. 30, l. 21 (paras. [0090] – [0096]). The digital processing system also includes a secure random access memory comprising a shared execution memory portion and a shared data memory portion, *See*, *e.g.*, Fig. 1, element 131; p. 10, ll. 12-19 (para. [0037]); Fig. 6, elements 620, 630; p. 27, l. 4 – p. 28, l. 14 (paras. [0085] – [0089]). In the digital processing system, to open a file in the plurality of files, the file metadata processing module is loaded in to the shared execution memory portion, and the file open operation is executed, wherein the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file and stores a cluster identifier for each cluster in the sequence in a buffer comprised in the shared data portion, wherein the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file. *See*, *e.g.*, Fig. 3, elements 340, 370; Fig. 7, elements 710, 730; p. 19, l. 11 – p. 22, l. 13. (paras. [0060] – [0068]; p. 28, l. 15 – p. 29, l. 21 (paras. [0090] – [0093]). In the digital processing system, to access the file, the file data processing module is loaded in the shared execution memory portion, wherein the data processing module overlays at least a portion of the metadata processing module, and the file access operation is executed, wherein the file access operation accesses a portion of data in the file using at least one cluster identifier stored in the buffer. *See*, *e.g.*, Fig. 7, elements 720, 740; p. 29, l. 30 – p. 30, l. 21 (paras. [0094] – [0096]).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 29-52 are unpatentable under 35 U.S.C. § 103(a) over U.S. Patent No. 6,567,887 ("Harmer") in view of U.S. Patent No. 6,604,170 ("Suzuki").

VII. ARGUMENT

A. The Harmer Reference

The Examiner rejected all of the appealed claims as being obvious, relying primarily on U.S. Patent No. 6,567,887 ("Harmer"). Harmer discloses a computer system with a disk drive and a disk caching mechanism that handles buffering of data being read from or written to the disk. *See, e.g.*, Harmer, Fig. 2; col. 3, l. 53 – col. 4, l. 39; Fig. 10; col. 7, ll. 27-47. Harmer further discloses using the caching mechanism to manage "the storage, lookup and updating of the file system information *directly out of system RAM*." Harmer, col. 4, ll. 34-38 (emphasis added). This file system information includes directory information and cluster information, *i.e.*, a file access table (FAT) and a directory structure, that is typically stored on a disk drive. *See*, *e.g.*, Harmer, col. 5, ll. 18-22; col.6, ll. 8-11. In other words, Harmer discloses using a caching mechanism that buffers reads and writes to/from the disk to also handle reads and writes of the FAT and the directory structure.

However, Harmer does not disclose that the caching mechanism buffers reads and writes of the FAT table and the directory structure, *i.e.*, that the caching mechanism accesses the disk at any time to read or write the FAT or the directory structure. To the contrary, Harmer specifically states that the disclosed system eliminates the need to read or write the directory structure or the FAT on the disk and that the caching mechanism accesses the directory structure and FAT directly from system RAM. *See* Harmer, col. 4, l. 40 - col. 5, l. 22. In addition, in Fig. 10, Harmer specifically states that the file system information, *e.g.*, the FAT and the directory structure, is "fixed in memory." Further, Harmer discloses that "[w]ith the present invention, it is no longer necessary to go to the drive to determine the location of the data to be read or written from the disk," *i.e.*, to read the FAT and/or directory structure from the disk. Also, note that in Fig. 2 of Harmer, the directory information 206, cluster information 204, and sector information 202 are depicted as being stored in the system RAM 50 and not on the mass storage peripheral device 56. In other words, Harmer discloses that the entire FAT and the entire directory structure are stored in the system RAM. Thus, when used to access the FAT and the directory structure, the caching mechanism is merely reading and writing the FAT and directory structure in RAM and does not access the disk.

B. Rejections under 35 U.S.C. § 103(a)

Claims 29-52 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,567,887 ("Harmer") in view of U.S. Patent No. 6,604,170 ("Suzuki"). Claim 29 is representative of this group of claims. This grouping is for purposes of this appeal only, and should not be construed to mean the patentability of any of the claims may be determined, in later actions before a court, based on the grouping. Rather, the presumption of 35 U.S.C. § 282 shall apply to each claim individually.

The question of obviousness under 35 U.S.C. § 103(a) is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art, (3) the level of skill in the art, and (4) where in evidence, so-called secondary considerations. *See Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966); *see also KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406-07 (2007) ("While the sequence of these questions might be reordered in any particular case, the [*Graham*] factors continue to define the inquiry that controls.").

"[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006), *cited with approval in KSR*, 550 U.S. at 418.

The Examiner has the initial burden to set forth the basis for any rejection so as to put the patent applicant on notice of the reasons why the applicant is not entitled to a patent on the claim scope that he seeks, *i.e.*, to establish a prima facie case for the rejection. *See In re Oetiker*, 977 F.2d 1443, 1445 (Fed. Cir.1992); *In re Piasecki*, 745 F.2d 1468, 1472 (Fed. Cir. 1984) (the initial burden of proof is on the USPTO "to produce the factual basis for its rejection of an application under sections 102 and 103") (quoting *In re Warner*, 379 F.2d 1011, 1016 (CCPA 1967)).

An appellant may attempt to overcome an examiner's obviousness rejection on appeal by submitting arguments and/or evidence to show that the examiner made an error in either (1) an underlying finding of fact upon which the final conclusion of obviousness was based, or (2) the reasoning used to reach the legal conclusion of obviousness. *See Kahn*, 441 F.3d at 985-86 ("On appeal to the Board, an applicant

can overcome a rejection by showing insufficient evidence of *prima facie* obviousness.") (quoting *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir.1998), *overruled in part on other grounds*, *KSR*, 550 U.S. at 422).

1. Claims 29-52

Claim 29 recites "opening the file using a file open operation comprised in a file metadata processing module loaded in a shared execution portion of the secure RAM, wherein … and stores a cluster identifier for each cluster in the sequence [of clusters allocated to the file] in a buffer comprised in a shared data portion of the secure RAM." The Examiner relies on Harmer, col., 6, ll. 18-22 and ll. 25-35 to allegedly teach these limitations, merely stating by way of explanation that these portions of Harmer disclose "clusters of files stored in FAT table which is cached in RAM." *See* Final Office Action dated October 22, 2009, p. 4.

The above cited limitations require that the storing of the cluster identifiers in the buffer in RAM occurs as a part of opening the file. As best Appellants can discern given the paucity of explanation provided by the Examiner, the Examiner is erroneously interpreting the disclosure of Harmer that file system data such as the directory structure and the FAT table is "cached" in RAM as somehow teaching the cited limitations. Appellants assert that the Examiner is improperly relying on selected portions of Harmer while ignoring the overall teachings of Harmer in making this rejection. Specifically, the portion of Harmer relied on by the Examiner states

> The file directory is found that corresponds to the request by reading it from the caching mechanism 210 which maintains that data in host RAM. Next, the cluster is found that corresponds to the file directory, and normally a request is sent to the drive to request data from the FAT table, and that information is read from the disk.

Harmer, col., 6, ll. 18-23. Taken out of context, this disclosure could be erroneously construed as disclosing that FAT information is being read from disk and buffered in memory as it does state that "normally a request is sent to the drive to request data from the FAT, and that information is read from the disk." However, when read in the context of the full disclosure of Harmer, such an interpretation is not supported. As was explained above, Harmer discloses that the entire directory structure and FAT are stored in RAM, so there is no need to access the disk to request data from the FAT during any file operation. Further, even the beginning of the paragraph including the

above quote states that "[w]ith the present invention, it is no longer necessary to go to the drive to determine the location of the data to be written or read from the disk." Harmer, col., 6, ll. 12-14. As was previously explained in Harmer, determining the location of data in a file requires both reading the directory structure and the FAT. *See* Harmer, col. 4, ll. 40-64.

Accordingly, Harmer cannot be read to disclose the above cited limitations of claim 29 and the Examiner has erred in an underlying finding of fact supporting the legal conclusion of obviousness. Thus, the Examiner has not established a prima facie case that the combination of Harmer and Suzuki renders claim 29 unpatentable.

Claim 29 further requires that "the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file." The Examiner asserts that Harmer, col. 6, ll. 14-18, discloses these limitations. *See* Final Office Action dated October 22, 2009, p. 4. This portion of Harmer merely states, in its entirety, that "[i]t is only necessary for the loadable device driver 66 to access or query the caching mechanism 210 in order to determine this information." A mere mention of accessing or querying a caching mechanism that manages file system information cannot possibly be read to disclose the specific storage requirements for cluster identifiers cited in claim 29. Further, as best Appellants can ascertain, nowhere in Harmer are the specifics of accessing or querying the caching mechanism disclosed.

Accordingly, Harmer cannot be read to disclose the above cited limitations of claim 29 and the Examiner has erred in an underlying finding of fact supporting the legal conclusion of obviousness. Thus, the Examiner has not established a prima facie case that the combination of Harmer and Suzuki renders claim 29 unpatentable.

Claim 29 also recites "accessing the file using a file access operation comprised in a file data processing module loaded in the shared execution portion, wherein the data processing module overlays at least a portion of the metadata processing module." The Examiner asserts that Harmer, col. 6, ll. 6-11, discloses these limitations, merely stating by way of explanation that this portion of Harmer discloses a "caching mechanism for FAT table" and "a cache inherently does not have the capacity to hold the entirety of a structure such as a hard drive." Final Office Action dated October 22, 2009, p. 4. This portion of Harmer states

> When a file is updated, this information is subsequently updated. Before data is actually read from the disk, the location of the data must

> be determined. The cache of the system, for example the FAT table, the directory structure, and the boot records, maintain the prior logical layout of the drive for the cache within the caching algorithm.

As best Appellant can tell given the paucity of explanation provided by the Examiner, the Examiner is erroneously attributing "inherent" functionality to the caching mechanism disclosed by Harmer that allows portions of the FAT table in RAM to be "overlaid" from disk during file operations. Such an attribution is in direct opposition to the teachings of Harmer. As was explained above, Harmer discloses that the entire FAT is stored in RAM and the caching mechanism only accesses the FAT in RAM.

Accordingly, Harmer cannot be read to disclose the above cited limitations of claim 29 and the Examiner has erred in an underlying finding of fact supporting the legal conclusion of obviousness. Thus, the Examiner has not established a prima facie case that the combination of Harmer and Suzuki renders claim 29 unpatentable.

Claim 29 also recites "opening the file using a file open operation comprised in a file metadata processing module loaded in a shared execution portion of the secure RAM, wherein the *file open operation* traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file." Emphasis added. The Examiner admits that Harmer does not disclose "the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file," and instead relies on Suzuki, col. 9, ll. 15-25 to allegedly disclose these limitations. *See* Final Office Action dated October 22, 2009, p. 4. While the portion of Suzuki relied on by the Examiner does disclose traversal of a file access table, the Examiner is conveniently ignoring the fact that this portion of Suzuki is describing "the read process of the found file," not the process of opening the file. Suzuki, col. 9, ll. 13-14. As one of ordinary skill in the art would know, a file open operation is used to find a file in a directory before a read operation is performed. In other words, when read in the appropriate context, the traversal of the FAT disclosed in Suzuki is performed as part of reading a file, not as part of opening a file as explicitly required by the cited limitations of claim 29. *See* Suzuki, col. 9, ll. 13-53.

Accordingly, Suzuki cannot be read to disclose the above cited limitations of claim 29 and the Examiner has erred in an underlying finding of fact supporting the legal conclusion of obviousness. Thus, the Examiner has not established a prima facie case that the combination of Harmer and Suzuki renders claim 29 unpatentable.

Further, the Examiner has not provided articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. The Examiner's stated reasoning is that "[i]t would have been obvious … to combine Harmer and Suzuki because using the step of 'the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file' would have given those skilled in the art the tools to improve the invention by allowing faster access to files by storing important structures in faster memory." Final Office Action dated October 22, 2009, p. 4. As was previously explained, Harmer discloses that the entire FAT is stored in RAM. Accordingly, those skilled in the art would know that the "important structures" are stored in "faster memory" in Harmer without any need for recourse to the teachings the Examiner attributes to Suzuki.

Based on the foregoing, the Examiner made errors in both the underlying findings of fact upon which the final conclusion of obviousness was based, and the reasoning used to reach the legal conclusion of obviousness. Accordingly, Appellants request that the rejection of this grouping be reversed, and the claims set for issue.

VIII. CONCLUSION

For the reasons stated above, Appellants respectfully submit that the Examiner erred in rejecting all pending claims. Appellants believe that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, please charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 relating to this matter to Deposit Account Number 20-0668, for Texas Instruments Incorporated.

Respectfully submitted,

/Ellen Baker Laws/

Ellen Baker Laws
Attorney for Appellants
Reg. No. 50272
713-937-8823

Texas Instruments Incorporated
P.O. Box 655474, MS 3999
Dallas, TX 75265
(972) 917-5287

IX. CLAIMS APPENDIX

1-28. (Canceled)

29. (Previously Presented) A method for accessing a file in a file system in a protected area comprised in secondary storage of a digital processing system comprising a secure random access memory (RAM), the method comprising:

opening the file using a file open operation comprised in a file metadata processing module loaded in a shared execution portion of the secure RAM, wherein the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file and stores a cluster identifier for each cluster in the sequence in a buffer comprised in a shared data portion of the secure RAM, wherein the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file; and

accessing the file using a file access operation comprised in a file data processing module loaded in the shared execution portion, wherein the data processing module overlays at least a portion of the metadata processing module, and wherein the file access operation accesses a portion of data in the file using at least one cluster identifier stored in the buffer.

30. (Previously Presented) The method of claim 1, wherein the file access operation comprises:

computing, based on a start index of the portion of data and the cluster size, an index into the buffer of a location of a cluster identifier of a cluster comprising a start of the data;

using the index to retrieve the cluster identifier from the buffer;

computing an offset within the cluster of the start of the data; and

issuing commands to access the data in the cluster starting at the offset.

31. (Previously Presented) The method of claim 1, wherein the cluster identifiers are stored sequentially in the buffer in cluster allocation order.

32. (Previously Presented) The method of claim 1, wherein the sequence of clusters consists of all clusters allocated to the file.

33. (Previously Presented) The method of claim 1, wherein opening the file and accessing the file are preformed in a secure mode of the digital processing system.

34. (Previously Presented) The method of claim 1, wherein each file in the file system has a same number of clusters and the buffer is of a size to store a cluster identifier for all clusters in a file.

35. (Previously Presented) The method of claim 1, wherein the buffer is overwritten each time a file in the file system is opened.

36. (Previously Presented) The method of claim 1, wherein the secondary storage is a secure digital card.

37. (Previously Presented) A machine readable non-volatile storage medium comprising executable instructions that, when executed by a processor of a digital processing system, cause performance of a method for accessing a file in a file system in a protected area comprised in secondary storage of the digital processing system, the method comprising:

opening the file using a file open operation comprised in a file metadata processing module loaded in a shared execution portion of a secure random access memory (RAM) comprised in the digital processing system, wherein the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file and stores a cluster identifier for each cluster in the sequence in a buffer comprised in a shared data portion of the secure RAM, wherein the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file; and

accessing the file using a file access operation comprised in a file data processing module loaded in the shared execution portion, wherein the data processing module overlays at least a portion of the metadata processing module, and wherein the file access operation accesses a portion of data in the file using at least one cluster identifier stored in the buffer.

38. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the file access operation comprises:

computing, based on a start index of the portion of data and the cluster size, an index into the buffer of a location of a cluster identifier of a cluster comprising a start of the data;

using the index to retrieve the cluster identifier from the buffer;

computing an offset within the cluster of the start of the data; and

issuing commands to access the data in the cluster starting at the offset.

39. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the cluster identifiers are stored sequentially in the buffer in cluster allocation order.

40. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the sequence of clusters consists of all clusters allocated to the file.

41. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein opening the file and accessing the file are preformed in a secure mode of the digital processing system.

42. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein each file in the file system has a same number of clusters and the buffer is of a size to store a cluster identifier for all clusters in a file.

43. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the buffer is overwritten each time a file in the file system is opened.

44. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the secondary storage is a secure digital card.

45. (Previously Presented) A digital processing system comprising:

a first secondary storage comprising a file system in a protected area, wherein the file system comprises a plurality of files;

a second secondary storage comprising a file metadata processing module comprising a file open operation and a file data processing module comprising a file access operation; and

a secure random access memory comprising a shared execution memory portion and a shared data memory portion,

wherein to open a file in the plurality of files,

the file metadata processing module is loaded in to the shared execution memory portion, and

the file open operation is executed, wherein the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file and stores a cluster identifier for each cluster in the sequence in a buffer comprised in the shared data portion, wherein the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file, and

wherein to access the file,

the file data processing module is loaded in the shared execution memory portion, wherein the data processing module overlays at least a portion of the metadata processing module, and

the file access operation is executed, wherein the file access operation accesses a portion of data in the file using at least one cluster identifier stored in the buffer.

46. (Previously Presented) The digital processing system of claim 45, wherein the file access operation:

computes, based on a start index of the portion of data and the cluster size, an index into the buffer of a location of a cluster identifier of a cluster comprising a start of the data;

uses the index to retrieve the cluster identifier from the buffer;

computes an offset within the cluster of the start of the data; and

issues commands to access the data in the cluster starting at the offset.

47. (Previously Presented) The digital processing system of claim 45, wherein the cluster identifiers are stored sequentially in the buffer in cluster allocation order.

48. (Previously Presented) The digital processing system of claim 45, wherein the sequence of clusters consists of all clusters allocated to the file.

49. (Previously Presented) The digital processing system of claim 45, wherein the file open operation and the file access operation are preformed in a secure mode of the digital processing system.

50. (Previously Presented) The digital processing system of claim 45, wherein each file in the file system has a same number of clusters and the buffer is of a size to store a cluster identifier for all clusters in a file.

51. (Previously Presented) The digital processing system of claim 45, wherein the buffer is overwritten each time a file in the file system is opened.

52. (Previously Presented) The digital processing system of claim 45, wherein the first secondary storage is a secure digital card.

X. EVIDENCE APPENDIX

None.

XI. RELATED PROCEEDINGS APPENDIX

None.